Real–Fake Speech Detection with CNNs, Embeddings, and Live Inference

Big Data Pipeline and Evaluation Report

Dragan Pavel

Course: Introduction to Big Data

Abstract

This report describes a practical audio deepfake detection pipeline built on the WaveFake/LJ Speech collection published on Hugging Face [1]. The system includes three complementary approaches: (i) a compact CNN operating on log-mel spectrograms, (ii) a pretrained wav2vec2 embedding baseline with a logistic regression classifier, and (iii) evaluation of an off-the-shelf deepfake detector from Hugging Face [2]. We emphasize data engineering choices (streaming, caching, and leakage-free splits), cross-generator evaluation, and a real-time microphone demo. Results show strong baseline performance for the CNN and embeddings, while the off-the-shelf detector fails to generalize to this dataset, highlighting the importance of transparent training data.

1 Introduction

Audio deepfakes created by modern TTS and voice conversion systems are increasingly realistic and pose risks for verification and media forensics. Large-scale benchmarks such as ASVspoof provide standardized evaluation protocols and illustrate the challenge of generalizing to unseen attacks [3, 4]. In research, models such as RawNet2 and AASIST demonstrate both end-to-end and spectro-temporal approaches to spoof detection [5, 6]. Our project complements these works with a reproducible pipeline that is also simple enough to implement.

2 Dataset and Big Data Considerations

We use the Hugging Face dataset `ajaykarthick/wavefake-audio`, which combines WaveFake synthetic speech with LJ Speech as real samples [7, 1]. The dataset is stored in Parquet shards, enabling efficient batch processing and streaming. For large-scale processing, the main bottlenecks are (i) feature extraction cost, (ii) disk I/O for cached features, and (iii) leakage through duplicate speakers or audio IDs. We therefore:

- split by `audio_id` to avoid leakage,
- support streaming and caching for scalable batch feature extraction,
- evaluate cross-generator generalization (WF6/WF7 held out).

3 Pipeline and Architecture

The pipeline follows three stages: (i) audio ingestion with streaming/caching, (ii) feature extraction to fixed-length log-mel spectrograms, and (iii) model inference and evaluation. Audio is resampled to 16 kHz and trimmed/padded to a 4 s window. For the CNN branch, we apply a compact 2D CNN with batch normalization and pooling. We implemented two CNN variants: a 4-block baseline and a 5-block extended model with dropout, as summarized in Table 1. For the embedding branch, we use mean-pooled wav2vec2 hidden states and train a logistic regression classifier.

Table 1: CNN variants used in this project.

Variant	Channels	Notes
Baseline CNN	[16, 32, 64, 128]	4 blocks, no dropout
Extended CNN (v2)	[16, 32, 64, 128, 256]	5 blocks, dropout 0.2

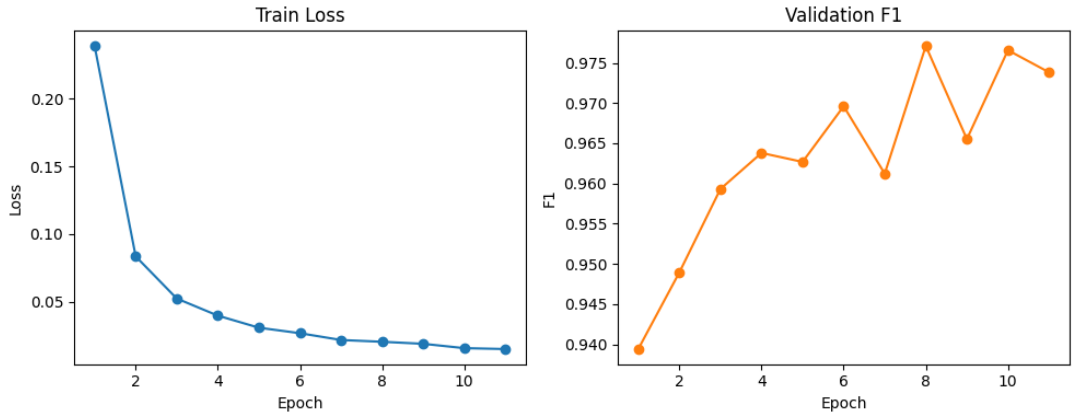

Figure 1: Training curves for the extended CNN baseline (v2).

The real-time demo processes a sliding window and applies exponential smoothing:

$$p_t = \alpha p_{t-1} + (1 - \alpha)\hat{p}_t, \tag{1}$$

where $\hat{p}_t$ is the raw fake probability and α controls stability.

4 Experiments

We evaluate four main configurations. Table 2 reports key parameters.

Table 2: Experiment settings.

Run	Input	Train Samples/Epoch	Notes
CNN baseline (v2)	log-mel (64)	60k	lower LR, no augmentation
CNN crossgen (v2)	log-mel (64)	60k	WF6/WF7 holdout, augmentation
Embeddings baseline	wav2vec2	20k	logistic regression
Embeddings crossgen	wav2vec2	20k	WF6/WF7 holdout

5 Results

Table 3 summarizes test performance. The extended CNN (v2) improves stability by lowering the learning rate and disabling augmentation for the baseline run, as reflected by the smoother loss/F1 trajectory in Figure 1. Embeddings slightly outperform the CNN in cross-generator generalization, while the CNN provides a strong baseline overall.

Table 3: Test results (accuracy and F1).

Run	**Accuracy**	**F1**
CNN baseline v2	0.963	0.979
CNN crossgen v2	0.879	0.916
Embeddings baseline	0.965	0.980
Embeddings crossgen	0.894	0.926

Figure 2 shows the cross-generator confusion matrix for the CNN v2 model. The model remains strong on fake recall but exhibits more false positives on real speech, illustrating the generalization challenge.

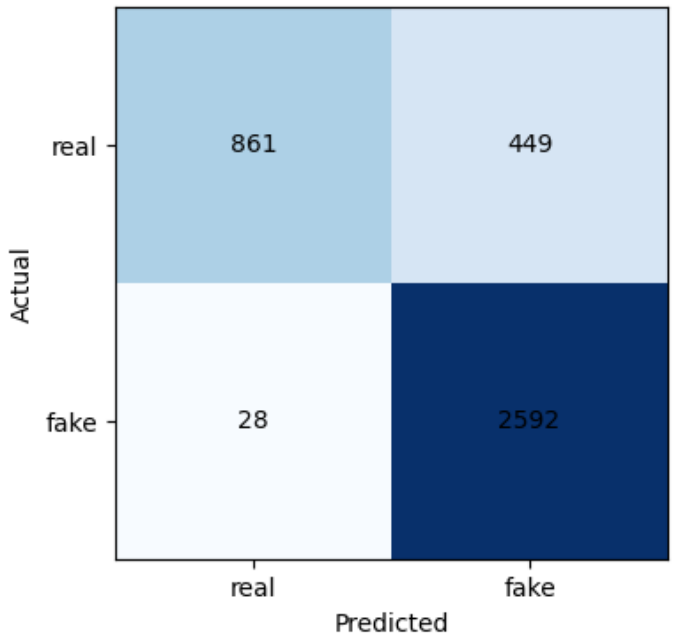

Figure 2: Cross-generator confusion matrix for CNN v2.

The off-the-shelf model [2] predicts nearly all samples as real on this dataset, producing near-zero F1. This suggests a mismatch between its training data and WaveFake and reinforces the need for transparent dataset documentation when comparing detectors.

6 Real-Time Demo

We implement a microphone-based demo that updates every 0.5 s using a 4 s window and Equation 1 for smoothing. A ring buffer enables near real-time inference, and a playback key allows users to verify

audio capture. Silence gating avoids spurious predictions when no speech is present. This demonstrates a stream-processing setting where latency and stability are critical.

7 Conclusion

We developed a full pipeline for audio deepfake detection with a focus on reproducibility and big data concerns. The CNN and wav2vec2 embedding baselines perform strongly, especially on cross-generator splits. The pretrained HF detector did not generalize, providing an important cautionary result. The real-time demo highlights practical deployment considerations such as smoothing and audio buffering. Future work could expand to more diverse datasets and calibration for improved real-time stability.

References

[1] A. Karthick, "Wavefake audio dataset (hugging face)," https://huggingface.co/datasets/ajaykarthick/wavefake-audio, 2023, accessed 2026-01-21.

[2] G. Stafford, "wav2vec2 deepfake voice detector (hugging face)," https://huggingface.co/garystafford/wav2vec2-deepfake-voice-detector, 2024, accessed 2026-01-21.

[3] J. Yamagishi, X. Wang, M. Todisco, M. Sahidullah, J. Patino, A. Nautsch *et al.*, "Asvspoof 2021: Accelerating progress in spoofed and deepfake speech detection," in *Proc. ASVspoof Challenge Workshop*, 2021, arXiv:2109.00537. [Online]. Available: https://arxiv.org/abs/2109.00537

[4] X. Wang, J. Yamagishi, M. Todisco, H. Delgado, A. Nautsch, N. Evans *et al.*, "Asvspoof 2019: A large-scale public database of synthesized, converted and replayed speech," *Computer Speech & Language*, 2020, arXiv:1911.01601. [Online]. Available: https://arxiv.org/abs/1911.01601

[5] H. Tak, J. Patino, M. Todisco, A. Nautsch, N. Evans, and A. Larcher, "End-to-end anti-spoofing with RawNet2," in *Proc. ICASSP*, 2021, arXiv:2011.01108. [Online]. Available: https://arxiv.org/abs/2011.01108

[6] J. weon Jung, H.-S. Heo, H. Tak, H. jin Shim, J. S. Chung, B.-J. Lee, H.-J. Yu, and N. Evans, "Aasist: Audio anti-spoofing using integrated spectro-temporal graph attention networks," *arXiv preprint arXiv:2110.01200*, 2021. [Online]. Available: https://arxiv.org/abs/2110.01200

[7] J. Frank and L. Schonherr, "Wavefake: A data set to facilitate audio deepfake detection," *arXiv preprint arXiv:2111.02813*, 2021. [Online]. Available: https://arxiv.org/abs/2111.02813